

First you have install the required packages in order use PHP and MySQL to create a web page. I am going to use NGINX as my webserver. Here are the commands to install the required packages:

```
pacman -S mariadb  
pacman -S php-fpm  
pacman -S nginx  
pacman -S vim
```

After you have downloaded and installed the packages, we are going to edit the configurations for our webserver, how it will work with PHP, and also where to read the files from to display on your website. Below is the command to access and edit your nginx configuration file:

```
vim /etc/nginx/nginx.conf
```

In the figure below you can see the outlined code in red. Those are the key parts that we will be changing within the file.

```
root@kenny ~]# vim /etc/nginx/nginx.conf
```

```
sendfile        on;
#tcp_nopush     on;

#keepalive_timeout 0;
keepalive_timeout 65;

#gzip on;

server {
    listen        8080;
    server_name   localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root      /usr/share/nginx/html;
        index      index.html index.htm;
    }

    #error_page 404              /404.html;

    # redirect server error pages to the static page /49x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root      /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ /\.php$ {
    #    proxy_pass http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    location ~ /\.php$ {
        fastcgi_pass   unix:/var/run/php-fpm/php-fpm.sock;
        fastcgi_index  index.php;
        root           /srv/http;
        include        fastcgi.conf;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny all;
    #}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#    listen        8000;
#    listen        somename:8080;
#    server_name   somename alias another.alias;
```

When you access you website, you enter in the ip address that your Raspberry Pi is connected to , you then add a forward slash “/” and enter the port number that it listens from. A simple command can find out your ip address for you if you type into the command line:

ip addr

Below is what the page should look like once you set it up.

